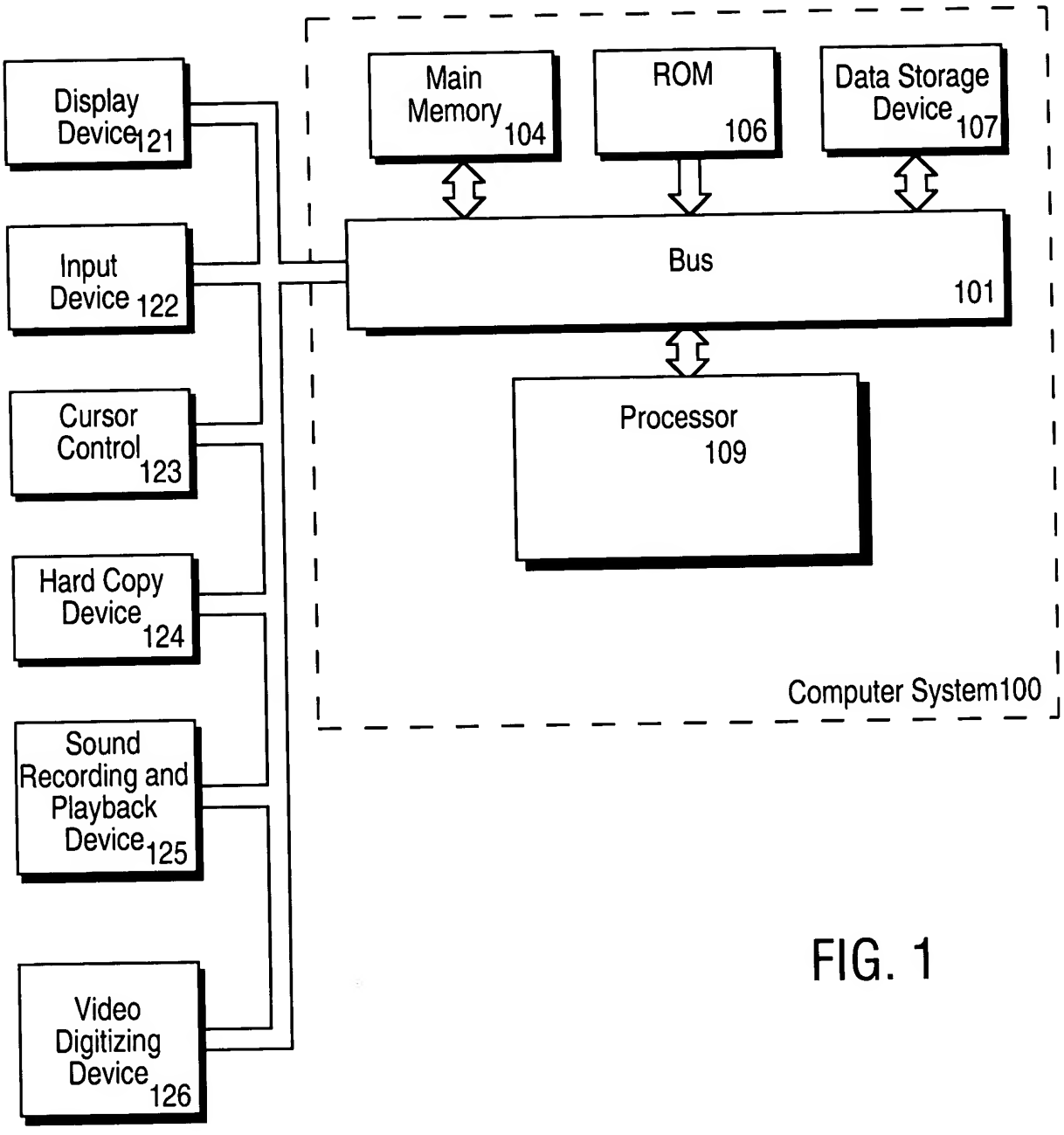
Display Device 121
Input Device 122
Cursor Control 123
Hard Copy Device 124
Sound Recording and Playback Device 125
Video Digitizing Device 126
Main Memory 104
ROM 106
Data Storage Device 107
Bus 101
Processor 109
Computer System 100

FIG. 1

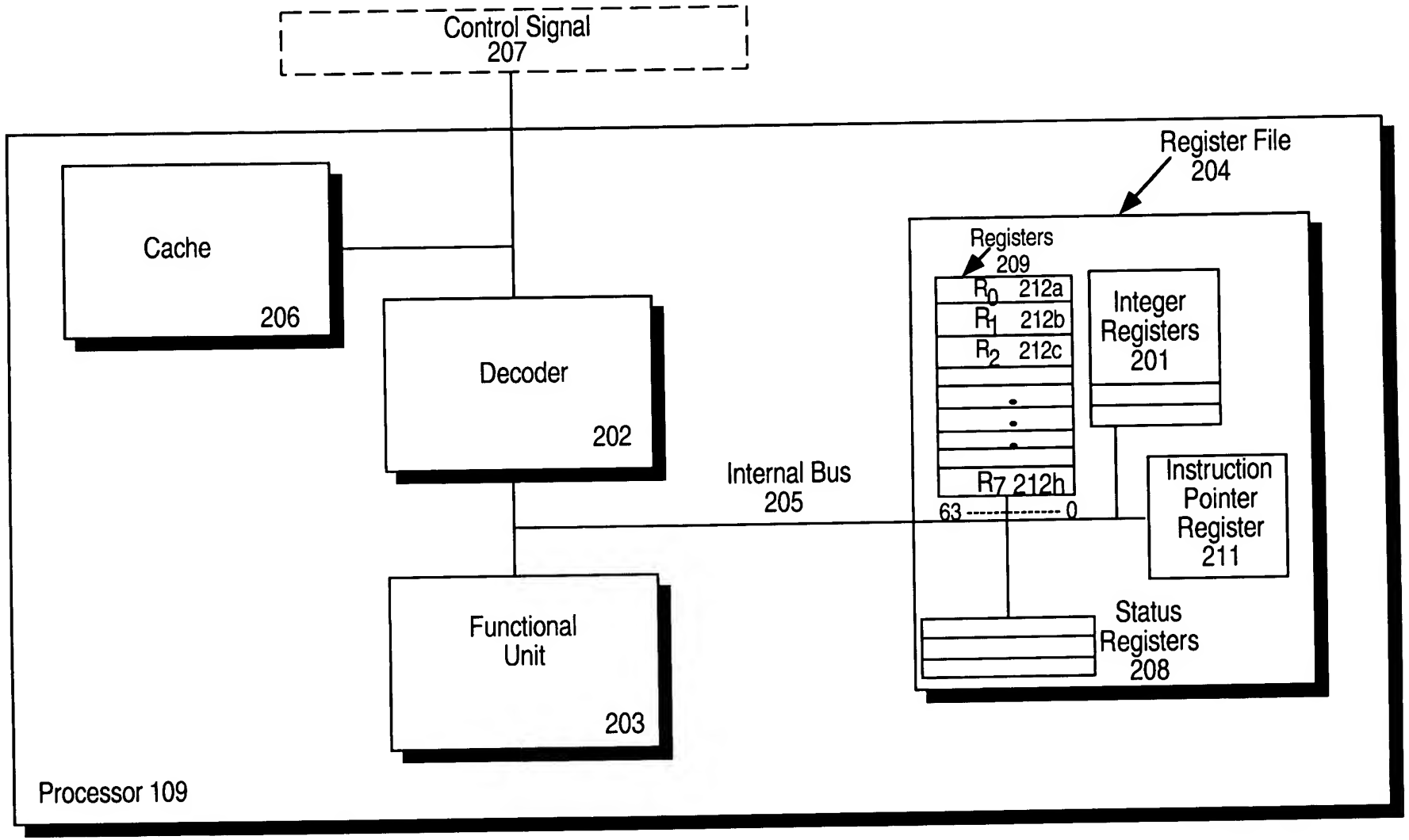
Control Signal
207
Register File
204
Cache
206
Registers
209
R_0 212a
R_1 212b
R_2 212c
R7 212h
Integer
Registers
201
Decoder
202
Internal Bus
205
63
0
Instruction
Pointer
Register
211
Functional
Unit
203
Status
Registers
208
Processor 109

FIG. 2

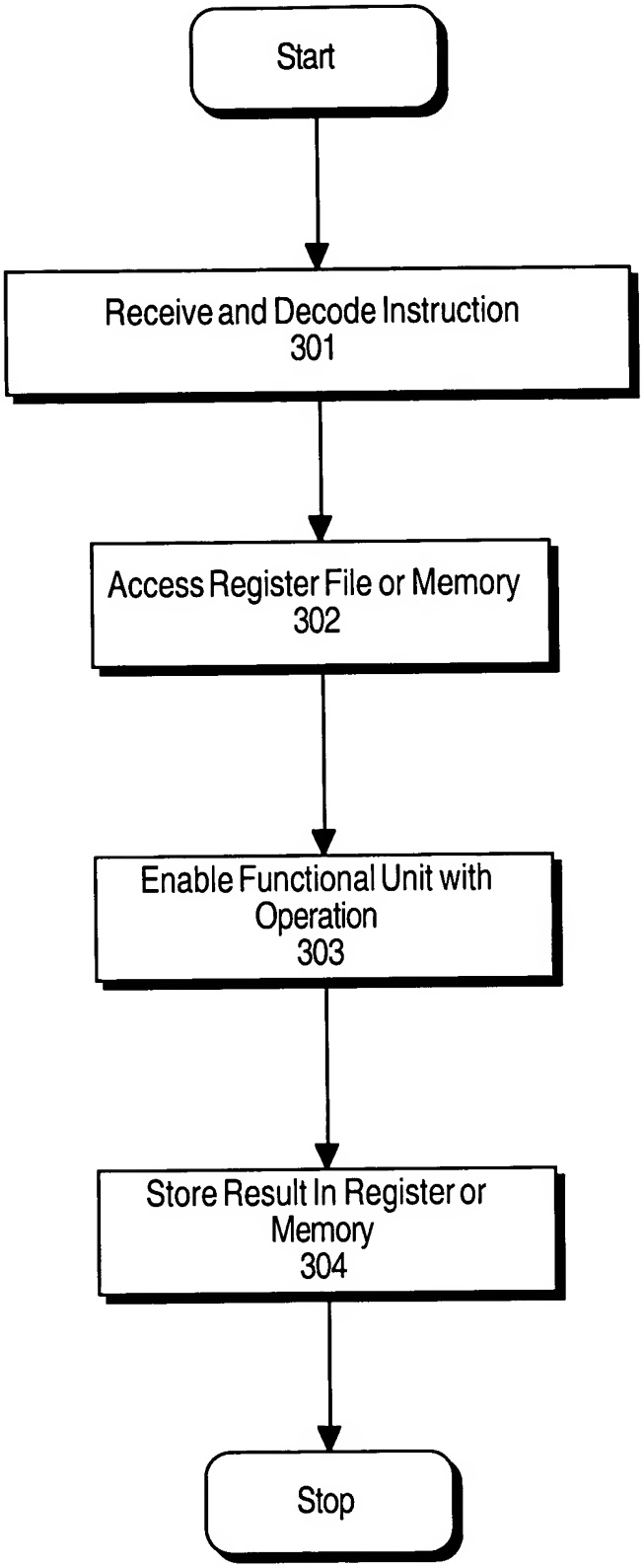
Start
Receive and Decode Instruction
301
Access Register File or Memory
302
Enable Functional Unit with
Operation
303
Store Result In Register or
Memory
304
Stop

FIG. 3

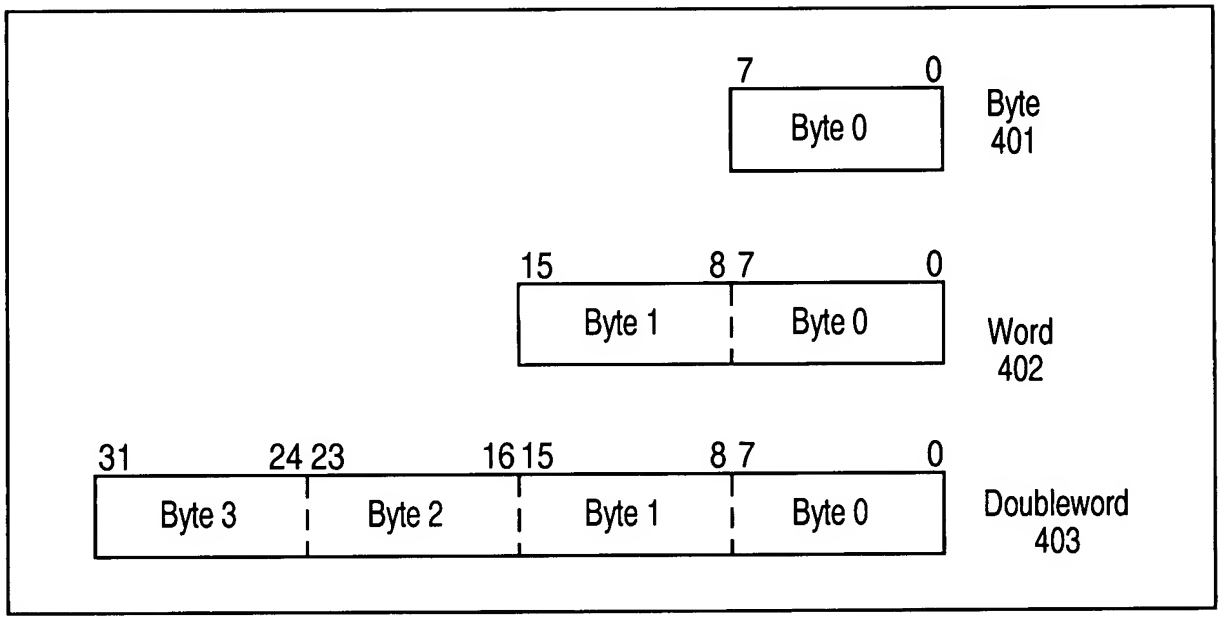
7
0
Byte 0
Byte
401
15
8 7
0
Byte 1
Byte 0
Word
402
31
24 23
16 15
8 7
0
Byte 3
Byte 2
Byte 1
Byte 0
Doubleword
403

FIG. 4A

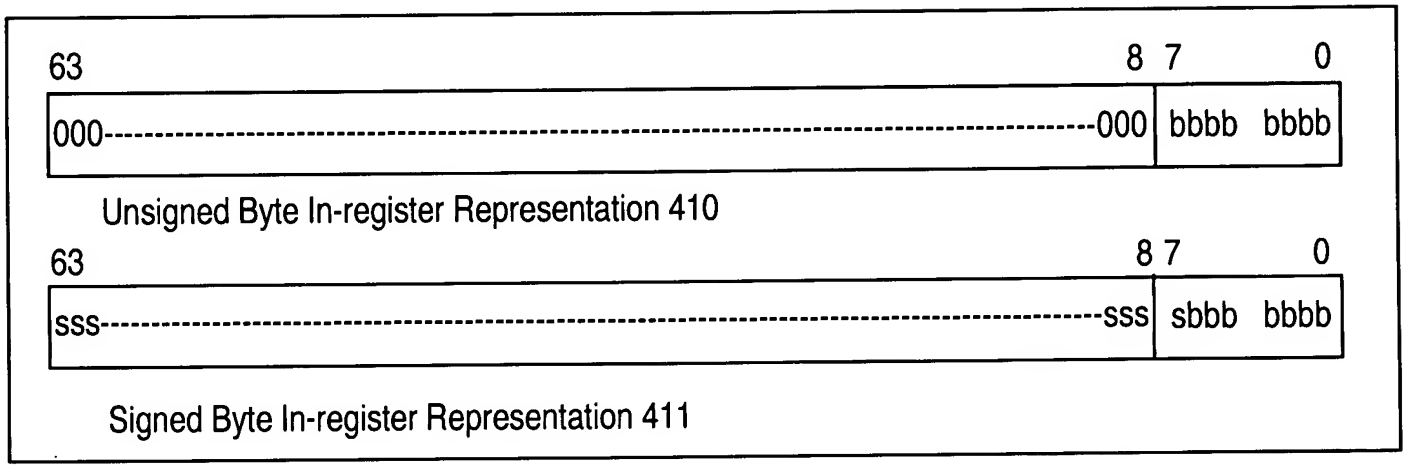
63
8 7
0
000----------000
bbbb bbbb
Unsigned Byte In-register Representation 410
63
8 7
0
sss----------sss
sbbb bbbb
Signed Byte In-register Representation 411

FIG. 4B

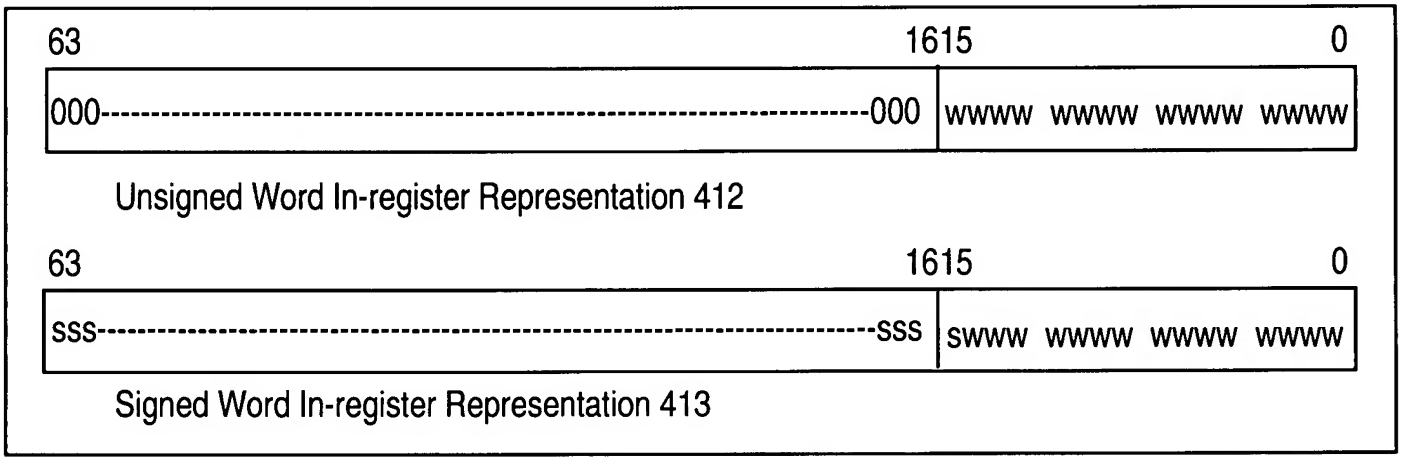
63
1615
0
000----------000
wwww wwww wwww wwww
Unsigned Word In-register Representation 412
63
1615
0
sss----------sss
swww wwww wwww wwww
Signed Word In-register Representation 413

FIG. 4C

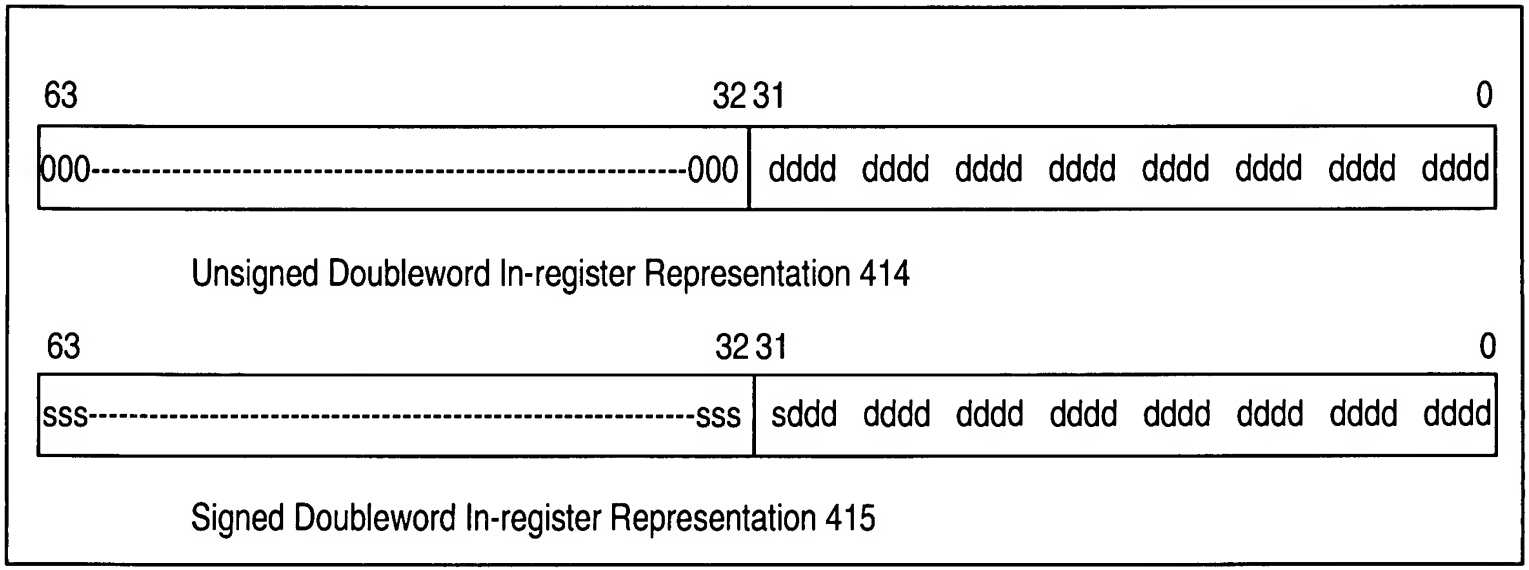
63
32 31
0
000----------000
dddd dddd dddd dddd dddd dddd dddd dddd
Unsigned Doubleword In-register Representation 414
63
32 31
0
sss----------sss
sddd dddd dddd dddd dddd dddd dddd dddd
Signed Doubleword In-register Representation 415

FIG. 4D

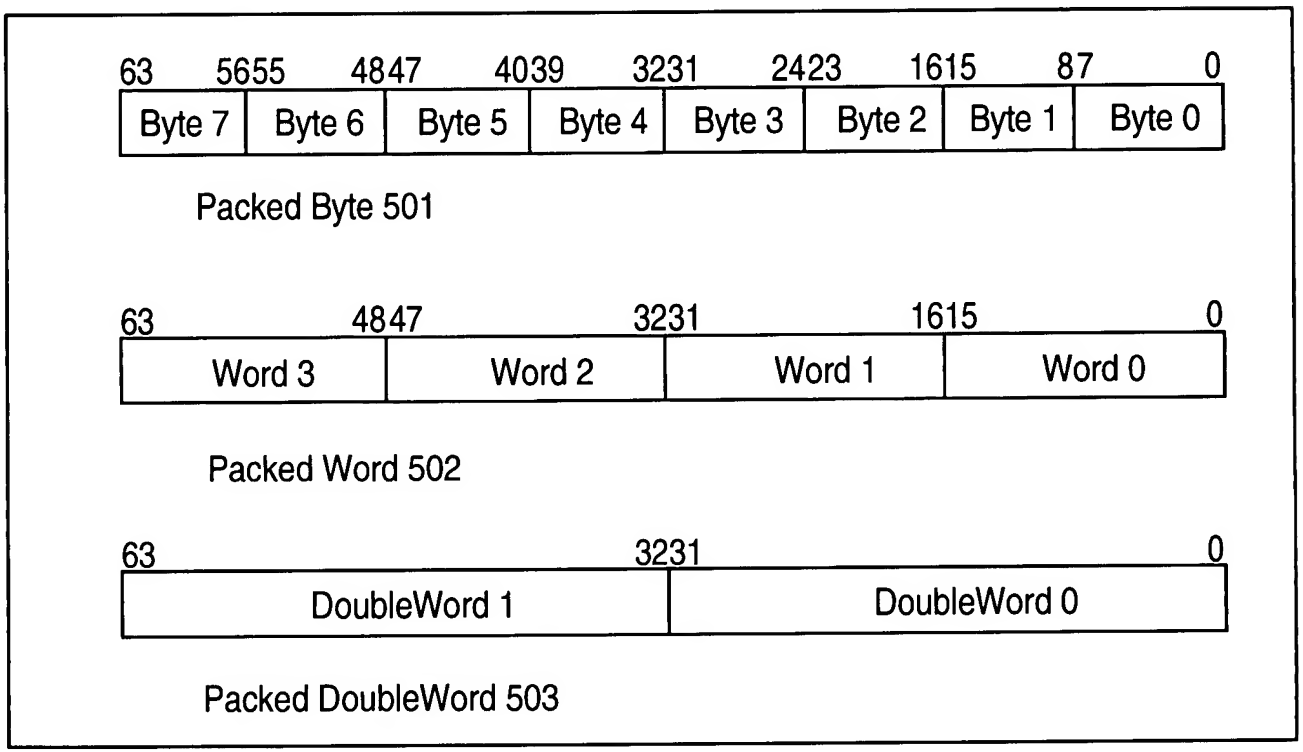
63 5655 4847 4039 3231 2423 1615 87 0
Byte 7
Byte 6
Byte 5
Byte 4
Byte 3
Byte 2
Byte 1
Byte 0
Packed Byte 501
63 4847 3231 1615 0
Word 3
Word 2
Word 1
Word 0
Packed Word 502
63 3231 0
DoubleWord 1
DoubleWord 0
Packed DoubleWord 503

FIG. 5A

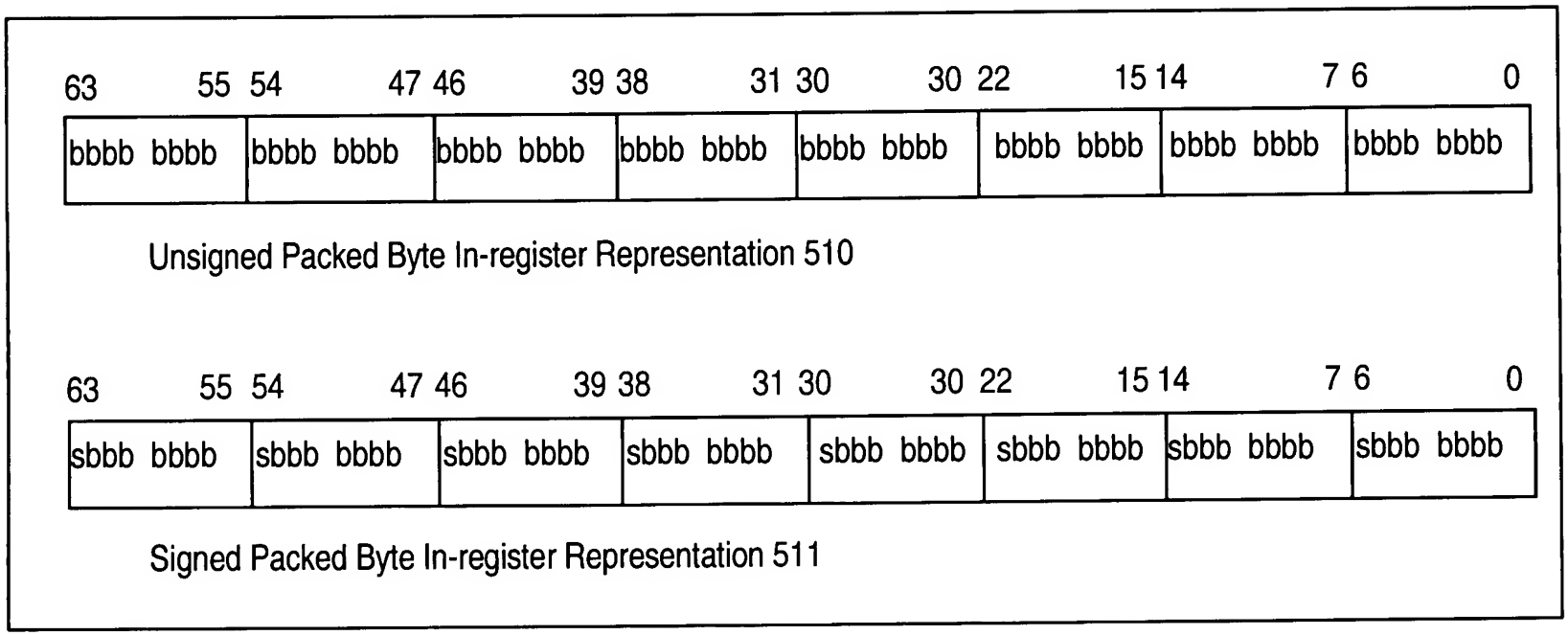
63 55 54 47 46 39 38 31 30 30 22 15 14 7 6 0
bbbb bbbb bbbb bbbb bbbb bbbb bbbb bbbb bbbb bbbb bbbb bbbb bbbb bbbb bbbb bbbb
Unsigned Packed Byte In-register Representation 510
63 55 54 47 46 39 38 31 30 30 22 15 14 7 6 0
sbbb bbbb sbbb bbbb sbbb bbbb sbbb bbbb sbbb bbbb sbbb bbbb sbbb bbbb sbbb bbbb
Signed Packed Byte In-register Representation 511

FIG. 5B

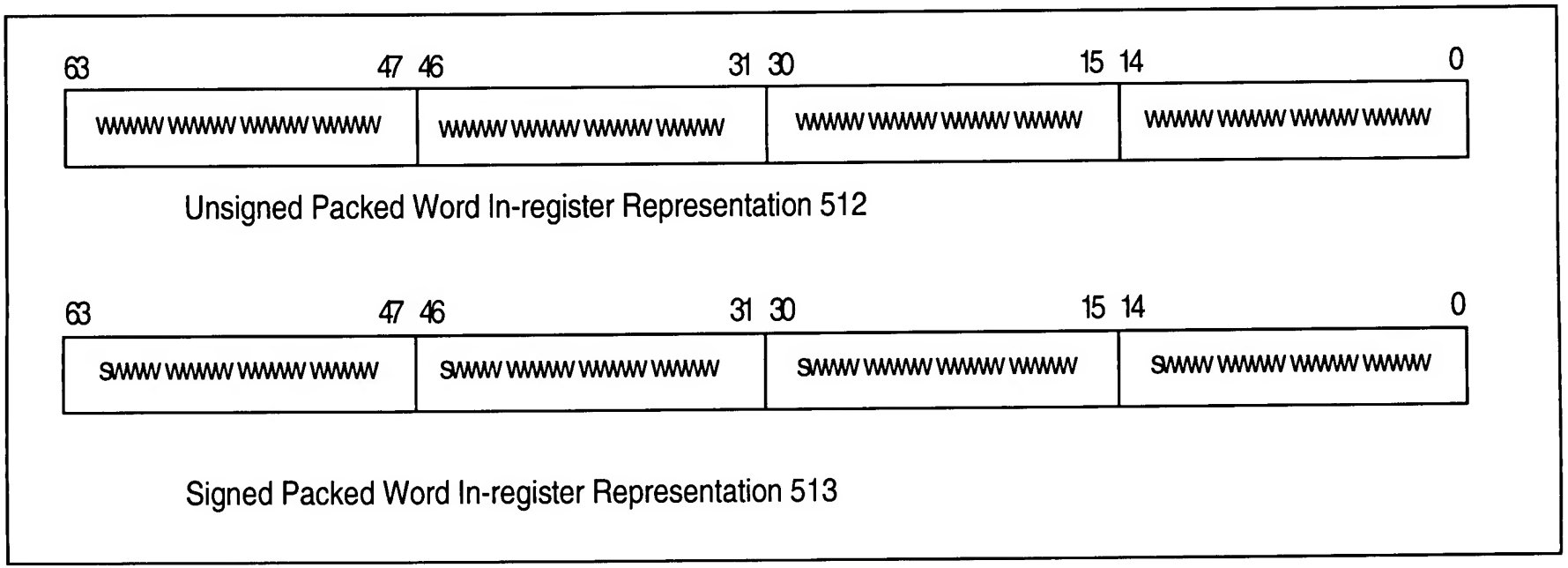
63
47 46
31 30
15 14
0
WWWW WWWW WWWW WWWW
WWWW WWWW WWWW WWWW
WWWW WWWW WWWW WWWW
WWWW WWWW WWWW WWWW
Unsigned Packed Word In-register Representation 512
63
47 46
31 30
15 14
0
SWWW WWWW WWWW WWWW
SWWW WWWW WWWW WWWW
SWWW WWWW WWWW WWWW
SWWW WWWW WWWW WWWW
Signed Packed Word In-register Representation 513

FIG. 5C

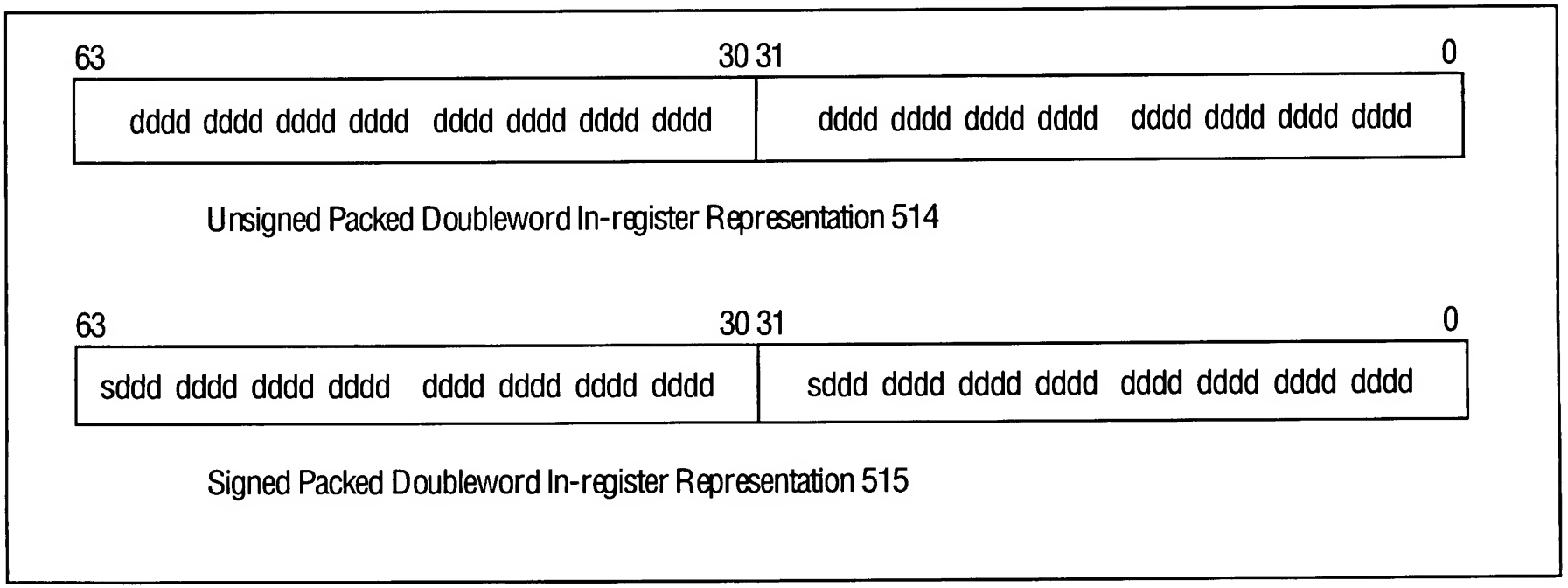
63
30 31
0
dddd dddd dddd dddd dddd dddd dddd dddd
dddd dddd dddd dddd dddd dddd dddd dddd
Unsigned Packed Doubleword In-register Representation 514
63
30 31
0
sddd dddd dddd dddd dddd dddd dddd dddd
sddd dddd dddd dddd dddd dddd dddd dddd
Signed Packed Doubleword In-register Representation 515

FIG. 5D

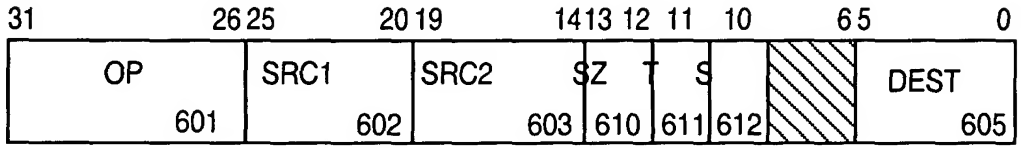
31
26 25
20 19
14 13 12
11
10
6 5
0
OP
601
SRC1
602
SRC2
603
SZ
610
T
611
S
612
DEST
605

FIG. 6A

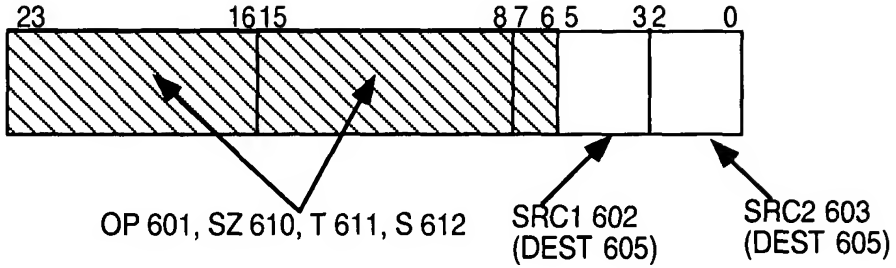
23
16 15
8 7 6 5
3 2
0
OP 601, SZ 610, T 611, S 612
SRC1 602
(DEST 605)
SRC2 603
(DEST 605)

FIG. 6B

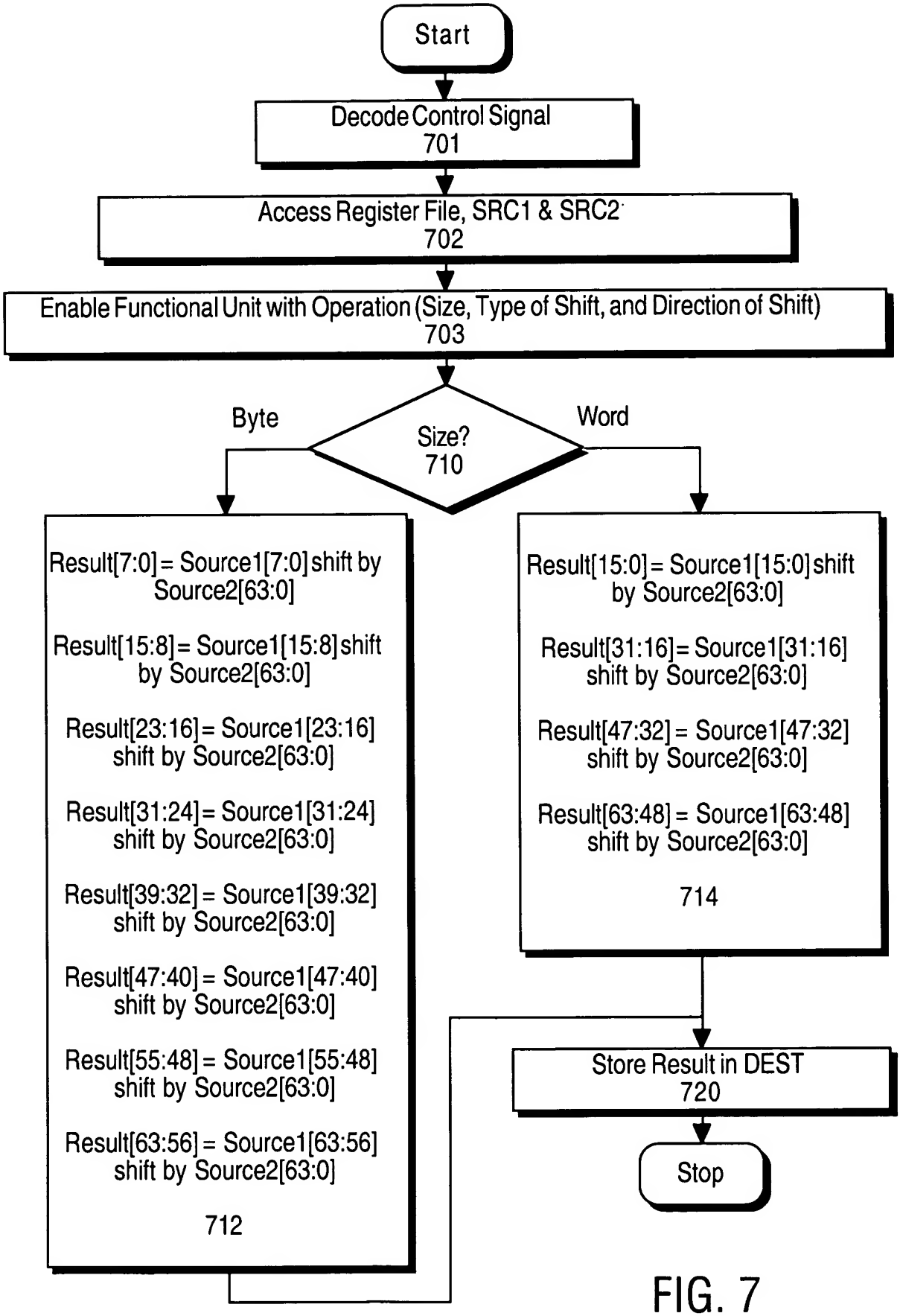
Start
Decode Control Signal
701
Access Register File, SRC1 & SRC2
702
Enable Functional Unit with Operation (Size, Type of Shift, and Direction of Shift)
703
Size?
710
Byte
Word
Result[7:0] = Source1[7:0] shift by Source2[63:0]
Result[15:8] = Source1[15:8] shift by Source2[63:0]
Result[23:16] = Source1[23:16] shift by Source2[63:0]
Result[31:24] = Source1[31:24] shift by Source2[63:0]
Result[39:32] = Source1[39:32] shift by Source2[63:0]
Result[47:40] = Source1[47:40] shift by Source2[63:0]
Result[55:48] = Source1[55:48] shift by Source2[63:0]
Result[63:56] = Source1[63:56] shift by Source2[63:0]
712
Result[15:0] = Source1[15:0] shift by Source2[63:0]
Result[31:16] = Source1[31:16] shift by Source2[63:0]
Result[47:32] = Source1[47:32] shift by Source2[63:0]
Result[63:48] = Source1[63:48] shift by Source2[63:0]
714
Store Result in DEST
720
Stop

FIG. 7

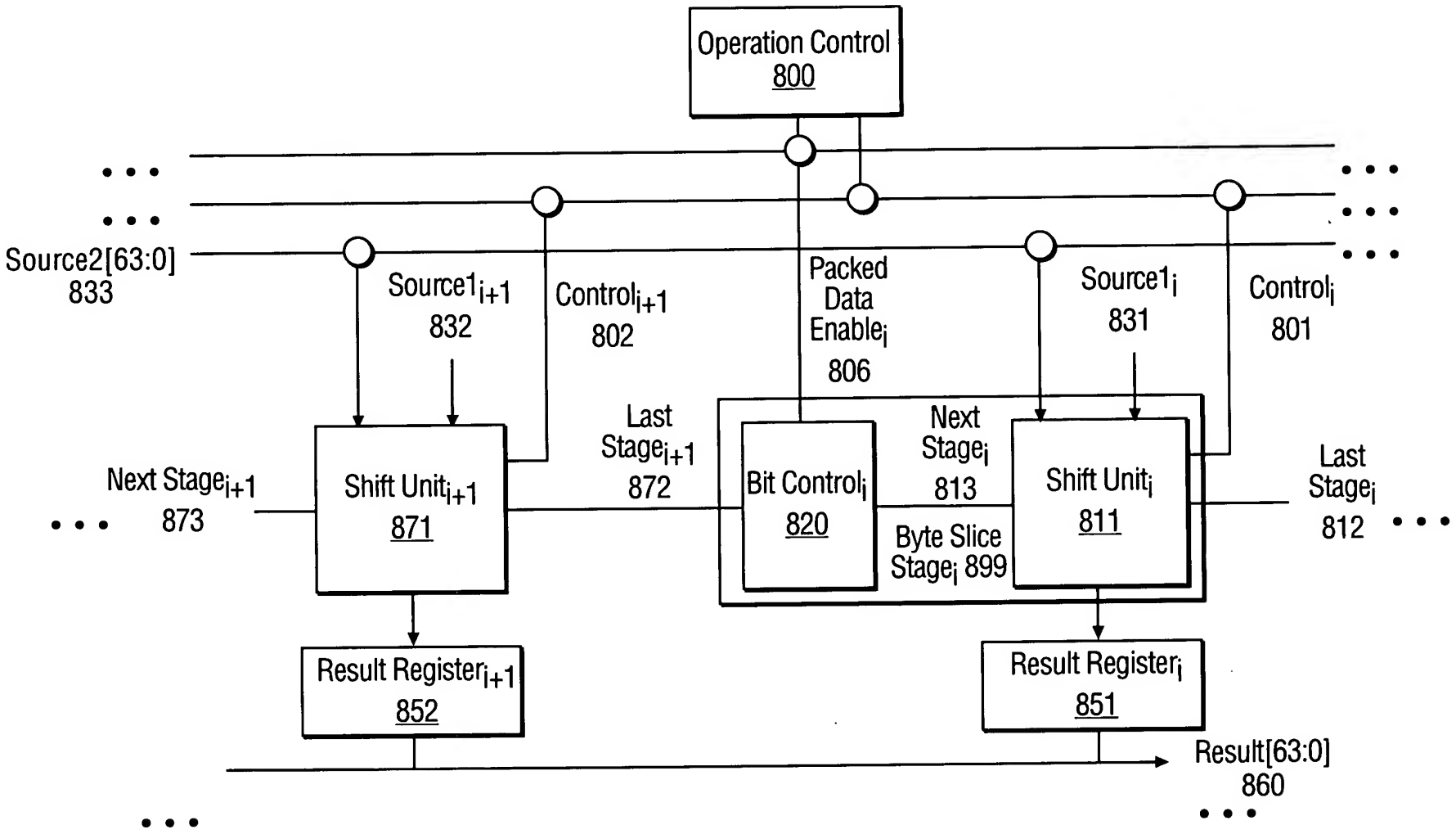
Operation Control
800
Source2[63:0]
833
Source1$_{i+1}$
832
Control$_{i+1}$
802
Packed
Data
Enable$_i$
806
Source1$_i$
831
Control$_i$
801
Next Stage$_{i+1}$
873
Shift Unit$_{i+1}$
871
Last
Stage$_{i+1}$
872
Bit Control$_i$
820
Next
Stage$_i$
813
Byte Slice
Stage$_i$ 899
Shift Unit$_i$
811
Last
Stage$_i$
812
Result Register$_{i+1}$
852
Result Register$_i$
851
Result[63:0]
860

FIG. 8

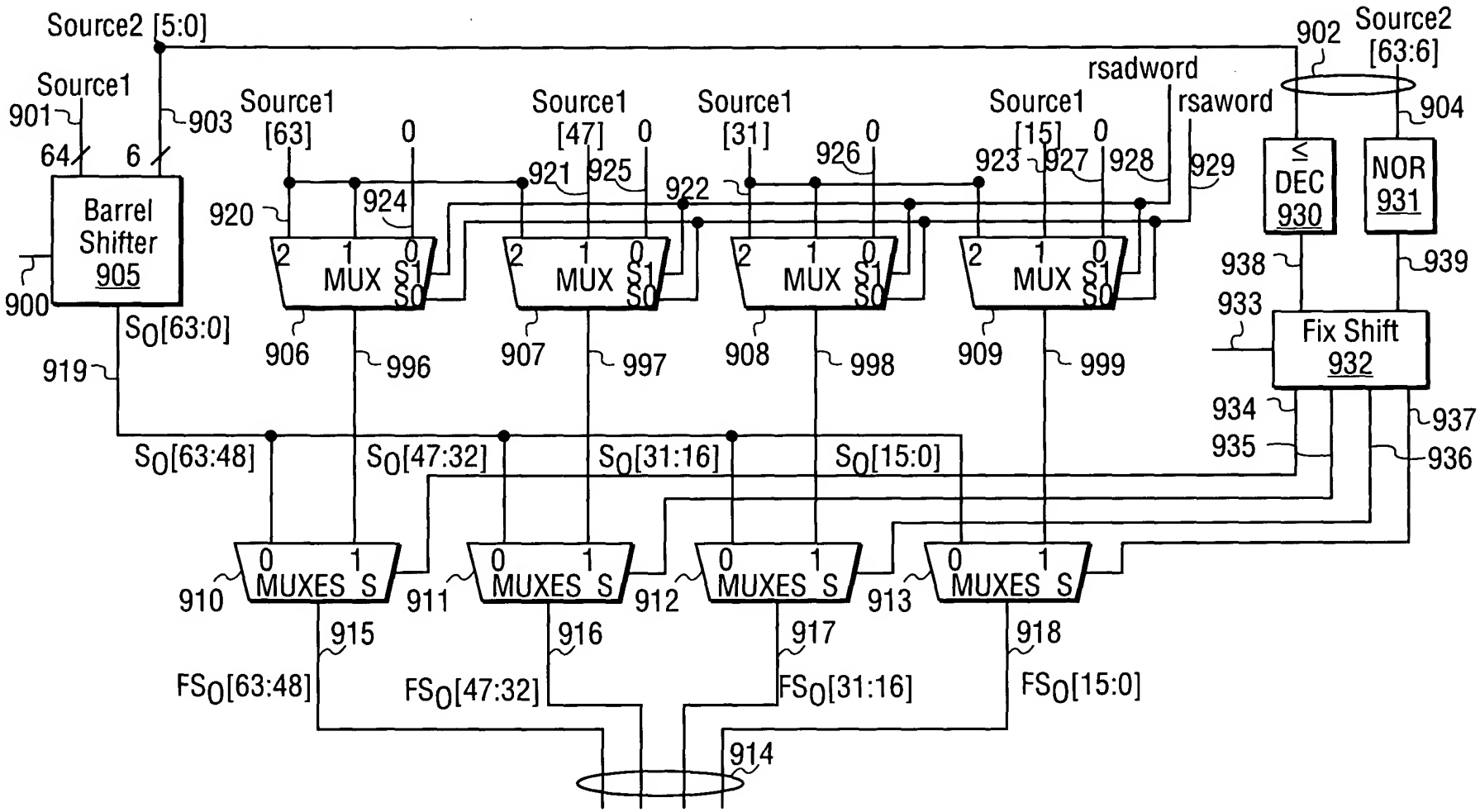
Source2 [5:0]
Source1
901
903
64
6
Barrel Shifter 905
900
S_O[63:0]
919
Source1 [63]
Source1 [47]
Source1 [31]
Source1 [15]
0
920
924
921
925
922
926
923
927
928
929
2 1 0 S1 S0 MUX
906
907
908
909
996
997
998
999
rsadword
rsaword
902
Source2 [63:6]
904
≤ DEC 930
NOR 931
938
939
933
Fix Shift 932
934
935
936
937
S_O[63:48]
S_O[47:32]
S_O[31:16]
S_O[15:0]
0 1 S MUXES
910
911
912
913
915
916
917
918
FS_O[63:48]
FS_O[47:32]
FS_O[31:16]
FS_O[15:0]
914

FIG. 9

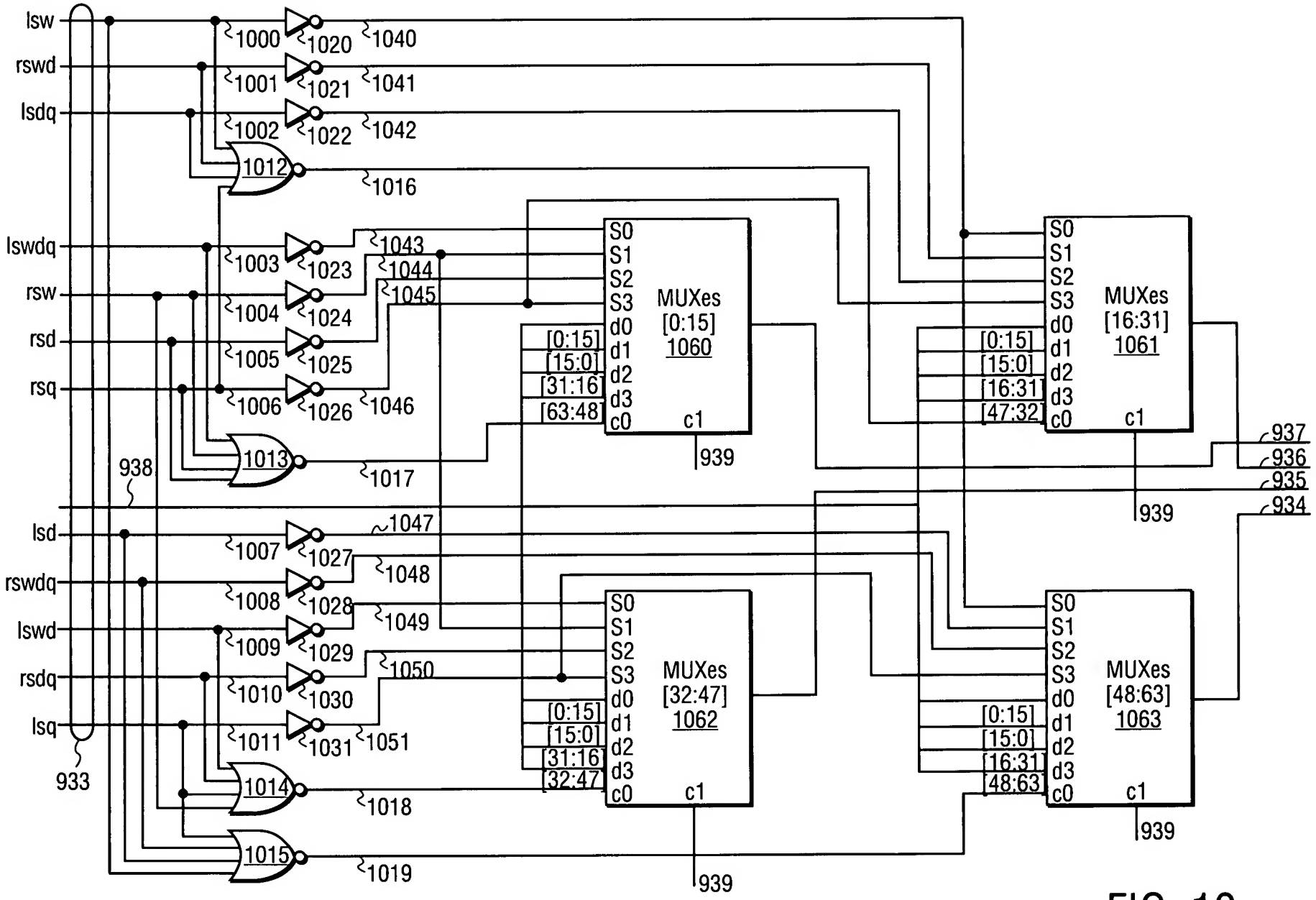
lsw
rswd
lsdq
lswdq
rsw
rsd
rsq
lsd
rswdq
lswd
rsdq
lsq
933
938
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
MUXes [0:15] 1060
MUXes [32:47] 1062
MUXes [16:31] 1061
MUXes [48:63] 1063
S0
S1
S2
S3
d0
d1
d2
d3
c0
c1
[0:15]
[15:0]
[31:16]
[63:48]
[32:47]
[16:31]
[47:32]
[48:63]
939
937
936
935
934

FIG. 10

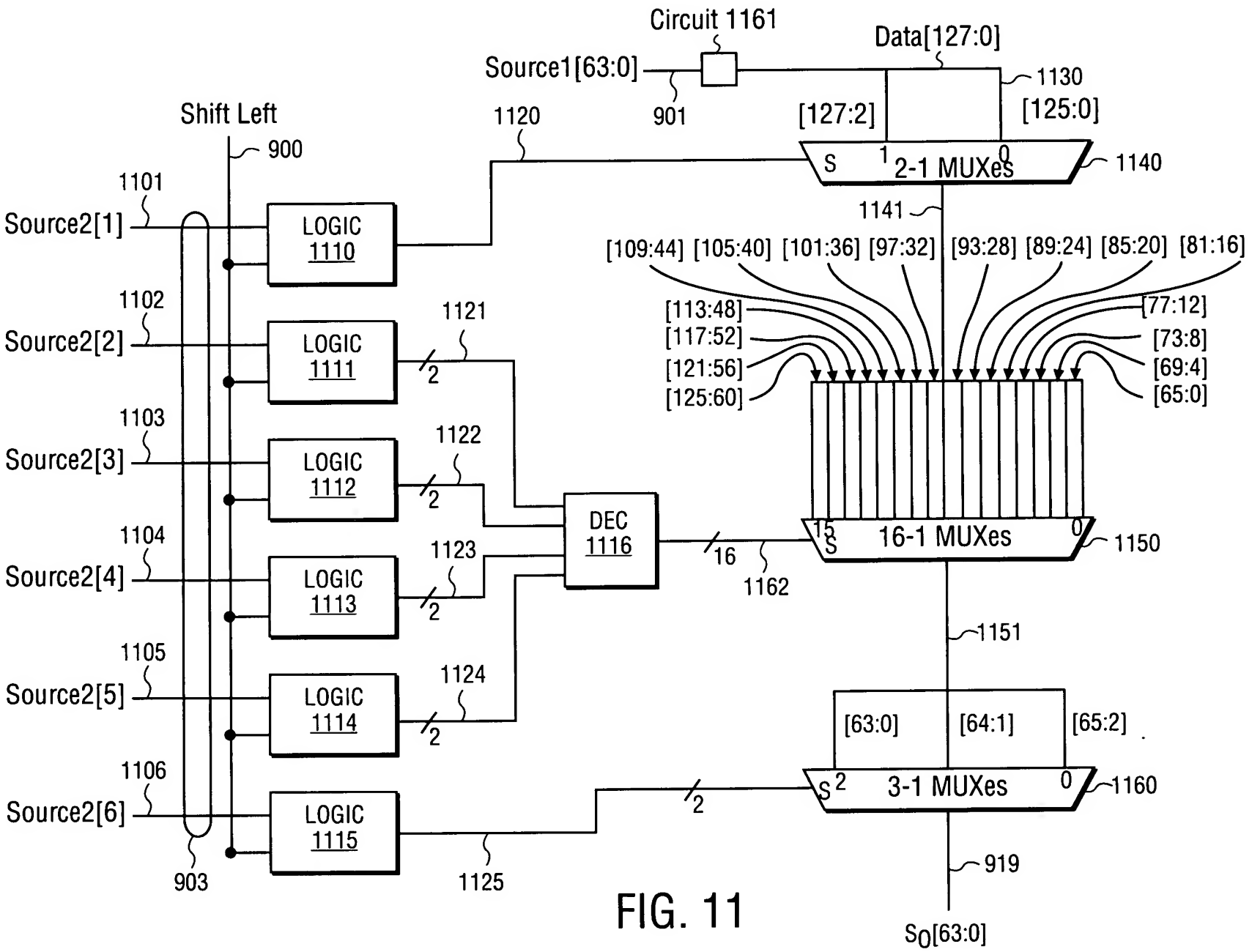
Circuit 1161
Source1[63:0]
901
Data[127:0]
1130
[127:2]
[125:0]
1120
Shift Left
900
S
1
0
2-1 MUXes
1140
1141
1101
Source2[1]
LOGIC 1110
[109:44] [105:40] [101:36] [97:32] [93:28] [89:24] [85:20] [81:16]
[113:48]
[117:52]
[121:56]
[125:60]
[77:12]
[73:8]
[69:4]
[65:0]
1102
Source2[2]
LOGIC 1111
1121
2
1103
Source2[3]
LOGIC 1112
1122
2
1104
Source2[4]
LOGIC 1113
1123
2
DEC 1116
16
1162
15
S
16-1 MUXes
0
1150
1151
1105
Source2[5]
LOGIC 1114
1124
2
[63:0]
[64:1]
[65:2]
1106
Source2[6]
LOGIC 1115
2
S
2
3-1 MUXes
0
1160
903
1125
919
S_O[63:0]

FIG. 11

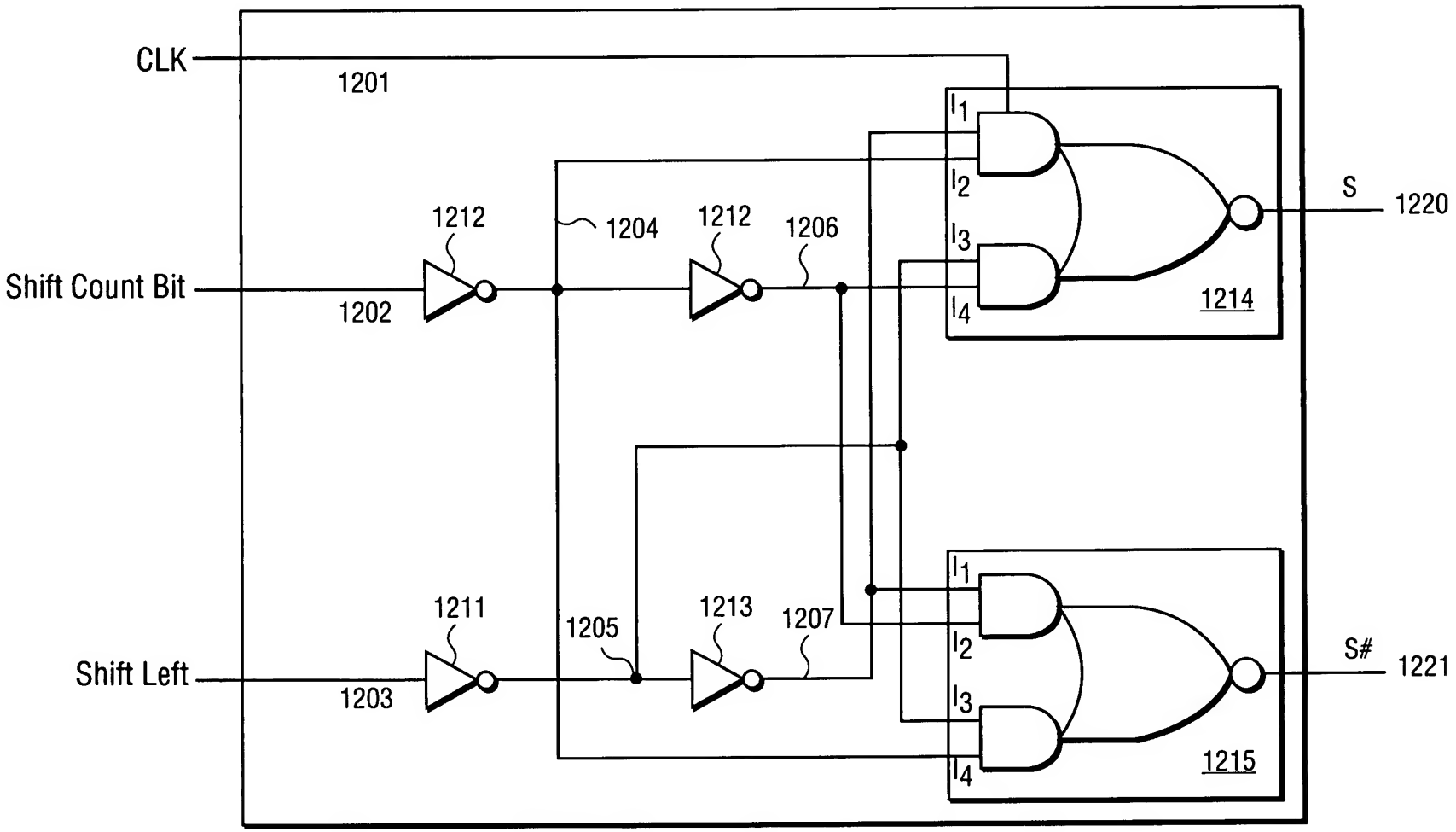
CLK
1201
Shift Count Bit
1202
1212
1204
1212
1206
Shift Left
1203
1211
1205
1213
1207
I1
I2
I3
I4
1214
S
1220
I1
I2
I3
I4
1215
S#
1221

FIG. 12

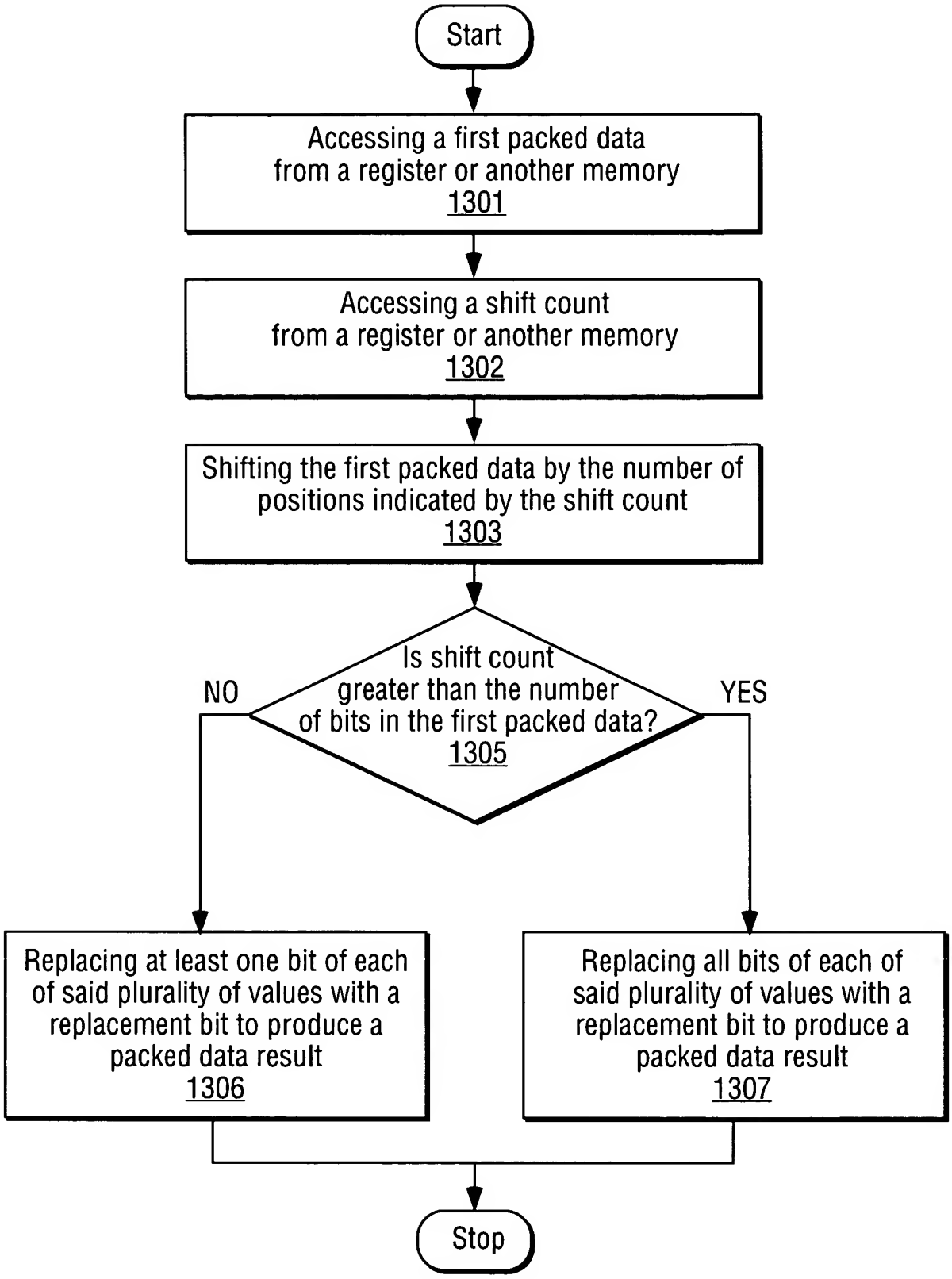
Start
Accessing a first packed data from a register or another memory 1301
Accessing a shift count from a register or another memory 1302
Shifting the first packed data by the number of positions indicated by the shift count 1303
Is shift count greater than the number of bits in the first packed data? 1305
NO
YES
Replacing at least one bit of each of said plurality of values with a replacement bit to produce a packed data result 1306
Replacing all bits of each of said plurality of values with a replacement bit to produce a packed data result 1307
Stop

FIG. 13